Présentation technique ApiDoc

Voici la documentation faite à l'aide d'un Bundle qui est NelmioApiDocBundle

1) A quoi sert NelmioApiDocBundle

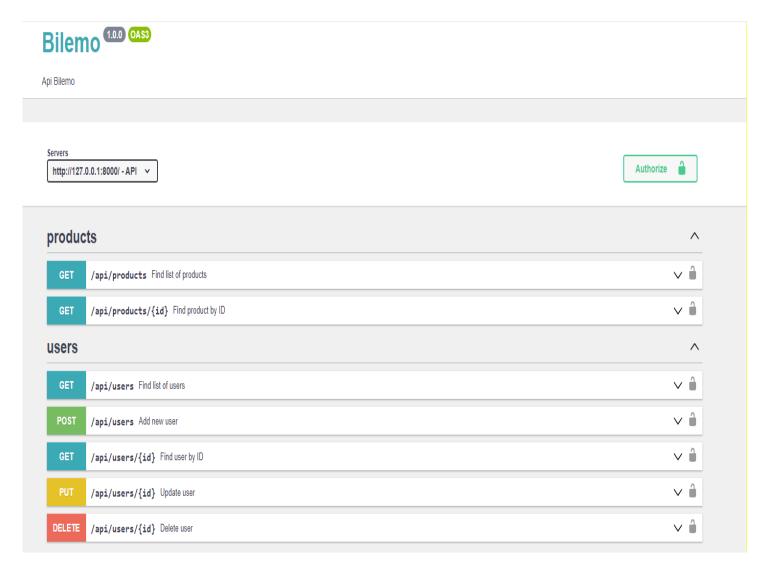
NelmioApiDocBundle est l'interface graphique permettant de visualiser et d'interagir avec ses API. Elle offre ainsi la possibilité de les tester facilement et de les rendre plus accessible pour le client.

2) Présentation de l'Apidoc

Voici le lien pour aller voir la Apidoc :

http://127.0.0.1:8000/api/doc

Vous arriverez sur la page d'accueil avec tous les liens urls qui sont présents pour votre API.



Avant de tester les liens, veuillez-vous procurer un token.

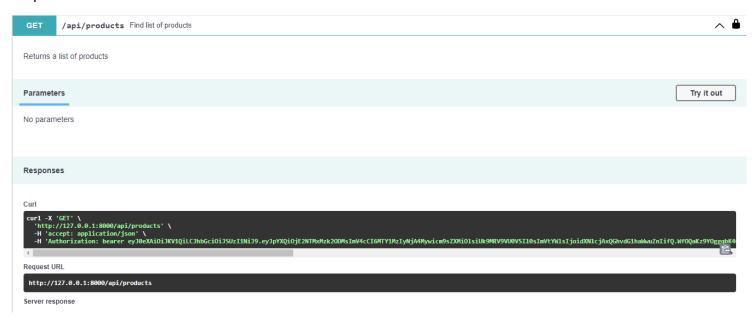
Cliquez sur « Authorize » pour vous authentifiez (ne pas oublier de mettre « Bearer » avant le token).

Nous allons voir chaque requête http:

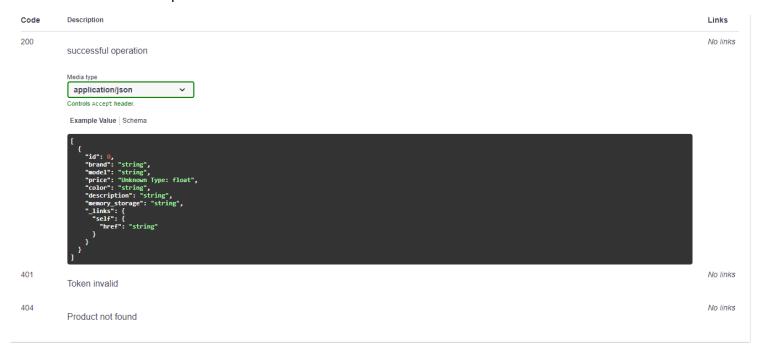
- Méthode : « GET », Route : « /api/products » :

Cette requête sert à récupérer la liste des utilisateurs.

Requête Curl et url:



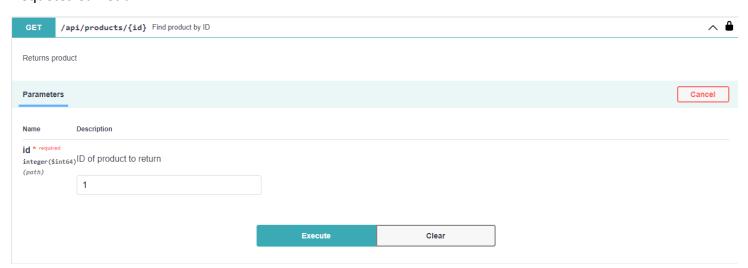
Reponse header et body:



- Méthode : « GET », route : « /api/products/{id} » :

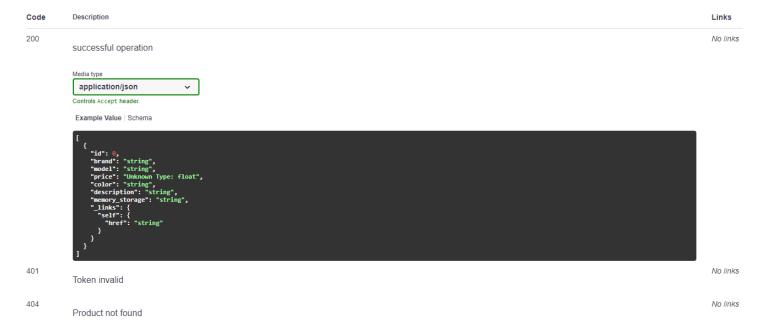
Cette requête sert à récupérer un produit grâce à l'id.

Requêtes Curl et url :



Réponse headers et body :

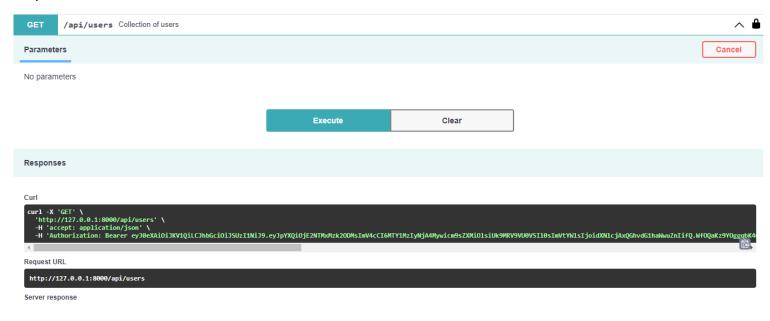
Le retour de code statut possible :



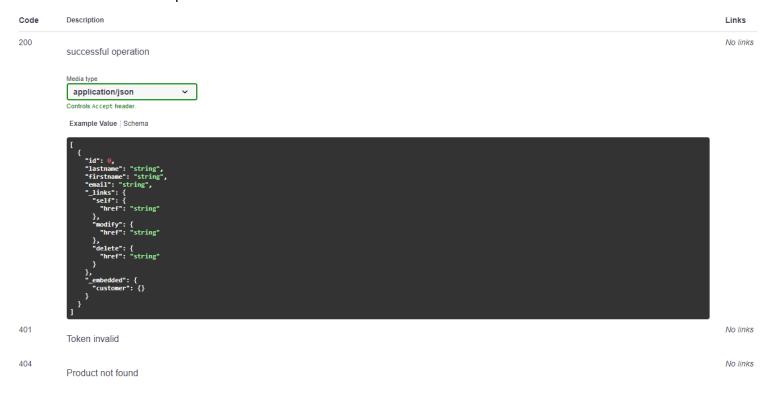
- Méthode : « GET », route : « /api/users » :

Cette requête sert à récupérer la liste des articles.

Requêtes Curl et url:



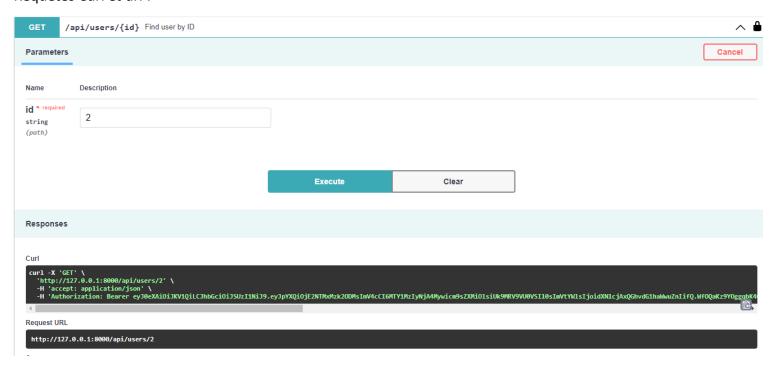
Réponse headers et body :



- Méthode : « GET », route : « /api/users/{id} » :

Cette requête http sert à récupérer un utilisateur grâce à l'id.

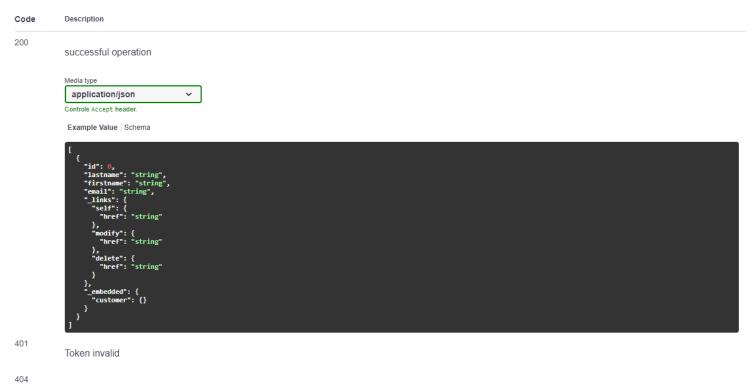
Requêtes Curl et url :



Réponse headers et body :

Le retour de code statut possible :

Product not found



- Méthode : « POST », route : « /api/users » :

Cette requête sert à créer un nouvel utilisateur

Requête body:

```
Request body required

{
    "lastname": "lastname",
    "firstname"; "firstname",
    "email": "email@hotmail.fr"

//

Execute
```

Requête Curl et url:

```
Curl -X 'POST' \
    'http://127.0.0.1:8000/api/users' \
    -H 'accept: application/json' \
    -H 'Authorization: Bearer ey10eXAi0iJKV1QiLCJhbGci0iJSUzI1NiJ9.eyJpYXQi0jE2NTMxMzk200MsImV4cCIGMTY1MzIyNjA4Mywicm9sZXMi01siUk9MRV9VU0VSI10sImVtYW1sIjoidXN1cjAxQGhvdG1haWwuZnIifQ.WfQQaKz9Y0ggqbK4-H 'Content-Type: application/json' \
    -d '{
        "lastname": "lastname",
        "email": "email@hotmail.fr"
};

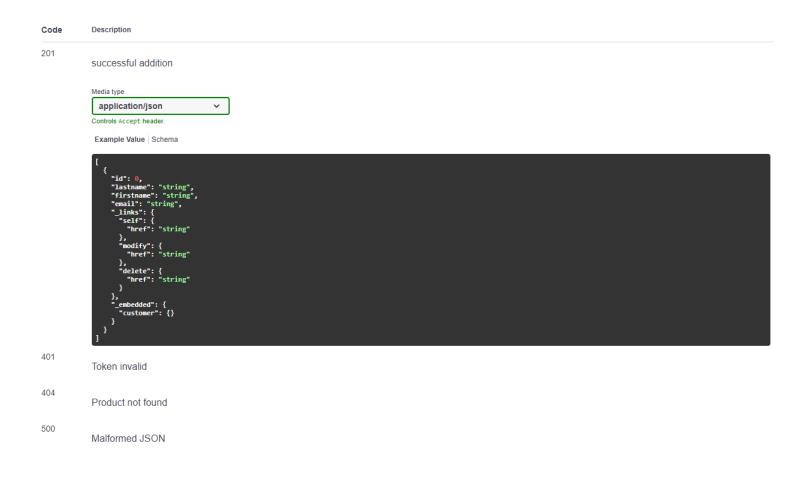
Request URL

http://127.0.0.1:8000/api/users

Server response
```

Réponse headers et body :

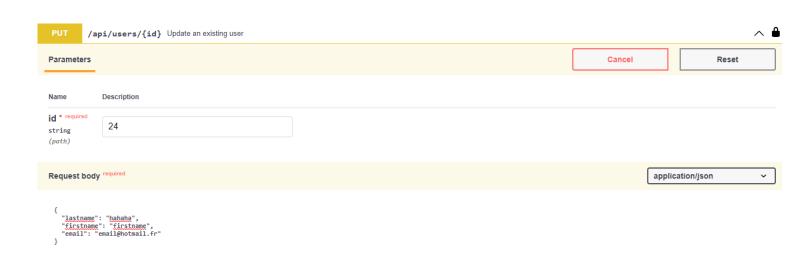
Le retour de code statut possible :



- Méthode : « PUT », route : « /api/users/{id} » :

Cette requête sert à modifier un utilisateur

Requête body:



Requête Curl et url:

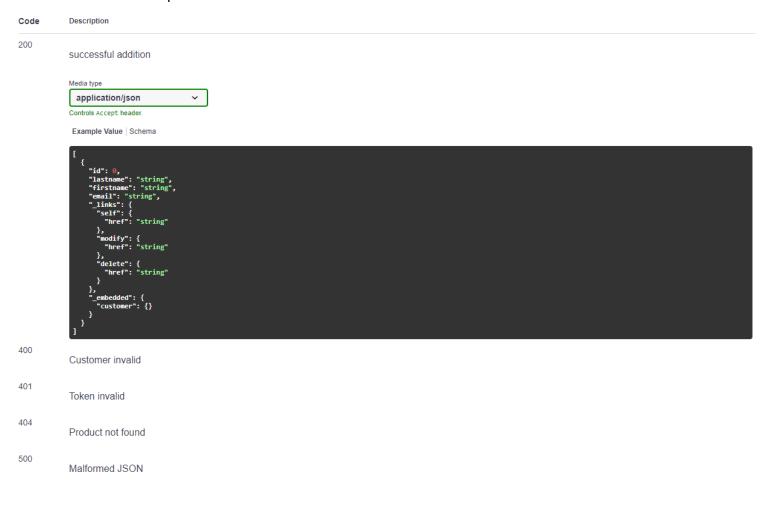
Réponse headers et body :

```
Code Details

Response body

\[
\begin{array}{l} \text{Response body} \\

\begin{array}{l} \text{Response body} \\
```



- Méthode : « POST », route : « /api/users » :

Cette requête sert à créer un nouvel utilisateur

Requête body:

Réponse requête url :

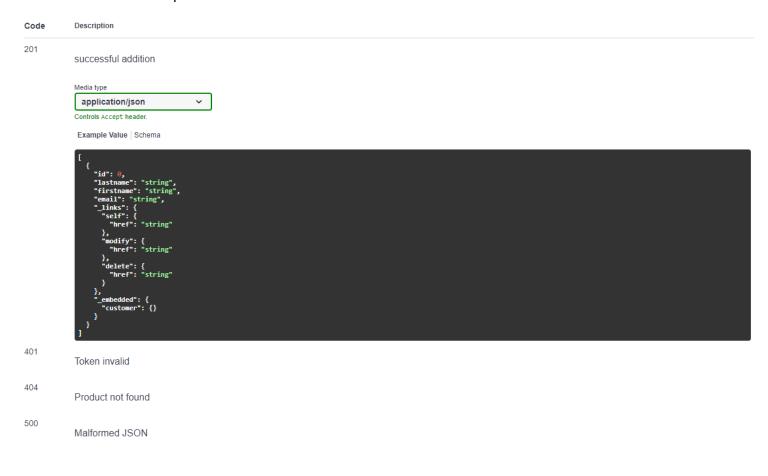
```
Curl -X 'POST' \
    'http://127.0.0.1:8000/api/users' \
    'http://127.0.0.1:8000/api/users' \
    'H' Authorization: Bearer eyJoeXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOjE2NTMxMzk2ODMsImV4cCIGMTY1MzJyNjA4Mywicm9sZXMiOlsiUk9MRV9VU0VSII0sImVtYWIsIjoidXNlcjAxQGhvdGlhaWwuZnIifQ.WfOQaKz9YOggqbK4-H 'Content-Type: application/json' \
    -d '{
        "lastname", "lastname",
        "email": "email@hotmail.fr"
}'

Request URL

http://127.0.0.1:8000/api/users

Server response
```

Réponse headers et body :



- Méthode : « DELETE », route : « /api/users/{id} » :

Cette requête sert à supprimer un utilisateur

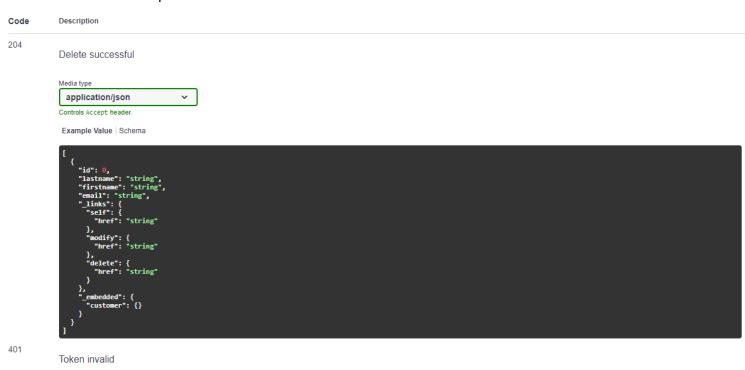
Requête Curl et url :

```
Curl -X 'DELETE' \
    'http://127.0.0.1:8000/api/users/24' \
    -H 'accept: application/json' \
    -H 'Authorization: Bearer eyJ0eXAi0iJKV1QiLCJhbGci0iJSUzIINiJ9.eyJpYXQi0jE2NTMbd/zk200MsImV4cCI6MTY1MzIyNjA4Mywicm9sZXMi01siUk9MRV9VU0VSI10sImVtYWlsIjoidXNlcjAxQGhvdGihaMhuZnIifQ.WfOQaKz9Y0gggbK4

Request URL

http://127.0.0.1:8000/api/users/24

Server response
```



404

Product not found